

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-099435

(43)Date of publication of application : 05.04.2002

(51)Int.Cl.

G06F 9/46

(21)Application number : 2000-286796

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 21.09.2000

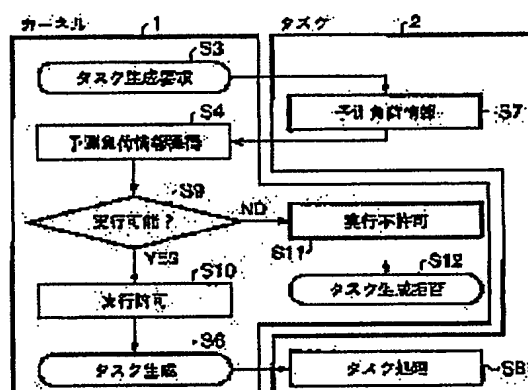
(72)Inventor : KOGURE HIROSHI

## (54) CONTROL METHOD OF MULTI-TASK AND INFORMATION PROCESSOR

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a control method of multi-task and an information processor using the method capable of deciding in advance whether it is possible to perform a task within a performing capacity of a processor and capable of avoiding multi-task processing beyond the capacity.

**SOLUTION:** In the method, when performing the task newly, at least the number of instruction code regarding to processing contents of the task is acquired from the task as a predicted load information using for a prediction of a load of the processor, so that the task decided to be performed within a reserve capacity of the processor based on the load information.



## LEGAL STATUS

[Date of request for examination] 24.07.2002

[Date of sending the examiner's decision of rejection] 28.03.2006

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

[0015] When a task generation (execution) request of a task is issued (step S3), a kernel 1 requests a task 2 to provide prediction load information. The task 2 provides the kernel 1 with the number of command code corresponding to a parameter of the task set at the present time as the prediction load information (step S7). The kernel 1, when obtaining the prediction load information from the task 2 (step S4), determines whether the task 2 can be executed within the available capacity of a processor (step S9) based on the prediction load information, an MIPS value of the processor, and the operation status of the processor at the present time (for example, in order to simplify the explanation, a ratio of the quantity of the command codes executable by the processor within a prescribed time period, which is calculated from the MIPS value, (processing capacity) to the total value of the quantity of the command codes of all tasks being executed by multi-task processing at the present time (or a measured value to a total value of the quantity of command codes presented from each task as prediction load information)). At that time, the kernel 1 can make a correction so that the quantity of the command codes, which is the prediction load information received from the task 2, depends on the hardware configuration of the processor. For example, an amount of overhead depending on the processor can be added to the quantity of the command code received from the task 2. Based on the predicted load information of the task 2 and the operation state of the processor, if the quantity of the command codes is the quantity of the command codes executable within the available capacity of the processor, it is determined that the task 2 can be executed. On the other hand, the quantity exceeds the quantity of the command codes executable within the available capacity of the processor, it is determined that the task 2 cannot be executed. If the task 2 is executable (step S10), it is generated (executed) (step S6). If the task 2 is not executable (step S9), the generation (execution) of the task 2 is rejected (step S12).

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号  
特開2002-99435  
(P2002-99435A)

(43)公開日 平成14年4月5日(2002.4.5)

(51)Int.Cl.<sup>7</sup>  
G 0 6 F 9/46

識別記号  
3 4 0

F I  
G 0 6 F 9/46

キーワード(参考)  
3 4 0 D 5 B 0 9 8

審査請求 未請求 請求項の数4 O L (全 6 頁)

(21)出願番号 特願2000-286796(P2000-286796)

(22)出願日 平成12年9月21日(2000.9.21)

(71)出願人 000003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(72)発明者 小暮 央

神奈川県川崎市幸区小向東芝町1番地 株

式会社東芝研究開発センター内

(74)代理人 100058479

弁理士 鈴江 武彦 (外6名)

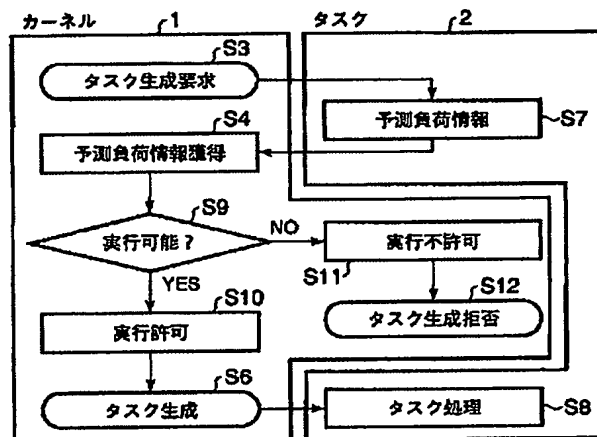
Fターム(参考) 5B098 GA02 GA04 GC09 GC10 GD02  
GD14

(54)【発明の名称】 マルチタスク制御方法および情報処理装置

(57)【要約】

【課題】プロセッサの処理能力内でそのタスクを実行可能か否かを予め判断でき、プロセッサの処理能力を超えるマルチタスク処理を回避することができるマルチタスク制御方法およびそれを用いた情報処理装置を提供する。

【解決手段】新たにタスクを実行させる際に、少なくともそのタスクの処理内容にかかる命令コードの数をプロセッサの負荷を予測するために用いる予測負荷情報として該タスクから獲得し、この予測負荷情報に基づきプロセッサの余力内で実行可能と判断したタスクを実行させる。



【0010】また、各タスクの予測負荷情報とは、例えば、そのタスクの処理内容にかかる命令コードの数（概算値）で、そのタスクを実行する際のプロセッサにかかる最悪の負荷量を推測する指標となる量である。そのタスクに予め設定されるパラメータ（処理内容を変更するようなパラメータ）が変化すれば、命令コードの数も変化する。この予測負荷情報も変化する。このようなパラメータとしては、タスクの処理内容をアプリケーションレベルで変更するためのもので、例えば、そのタスクの処理対象データに対し、サンプリング周波数を変えたり、フィルタ係数を変えたりするパラメータがある。この予測負荷情報は、複数のパラメータがあれば、そのパラメータに対応してタスク自身が保持するか、あるいは、計算するものである。その場合、タスクは、予測負荷情報をパラメータに対応付けて保持あるいは計算する部分とタスク本来の処理を行う部分に分かれているものとする。以下の実施形態では、説明の簡単のため、タスクが1または複数のパラメータのそれぞれに自身の命令コードの数（あるいは、命令コードの数にプロセッサに依存するオーバーヘッド分を加算したもの）を対応付けて予め保持するものとする。

【0011】あるタスクのタスク生成（実行）要求があった場合（ステップS3）、カーネル1はタスク2に予測負荷情報の提供を依頼する。タスク2は該タスクの現時点に設定されているパラメータに対応する命令コードの数を予測負荷情報として（ステップS7）、カーネル1に渡す。カーネル1はタスク2から予測負荷情報を獲得すると（ステップS4）、その予測負荷情報とプロセッサのMIPS（Million Instructions Per Second）値とプロセッサの現時点での稼働状況（例えば、ここでは、説明の簡単のため、MIPS値から求まるプロセッサが所定時間内に実行可能な命令コードの数（処理能力）に対する、現時点でマルチタスク処理で実行されている全てのタスクの命令コードの数の合計値（実測値を用いる場合と、予測負荷情報として各タスクから提示された命令コード数の合計値であってもよい）の割合）に基づいて、プロセッサの余力内でのタスク2の実行可能性を判断し、タスク2の優先順位を決定する（ステップS5）。このとき、カーネル1は、タスク2から受け取った予測不可情報としての命令コードの数をプロセッサのハードウェア構成に依存する修正を行ってもよい。例えば、タスク2から受け取った命令コードの数にプロセッサに依存するオーバーヘッド分を加算してもよい。

【0012】タスク2の予測負荷情報とプロセッサの稼働状況とから、タスク2の命令コードの数がプロセッサの余力内で実行可能な命令コードの数であるとき、実行可能と判断し、高い優先順位をタスク2に与え、一方、

位をタスク2に与える。その後、優先順位に基づいてタスク2を生成（実行）し（ステップS6）、タスク2はタスク処理を行う（ステップS8）。

【0013】例えば、図4のように、プロセッサ20がエンコーダ・タスク21とデコーダ・タスク22を実行中だとして、フィルタA・タスク23の生成要求が発生した場合（ステップS25）を考える。カーネル1はフィルタA・タスク23から予測負荷情報を知らされ、カーネルは予測負荷情報に基づいてフィルタA・タスク23の優先度を「10」と決定する。一方、フィルタB・タスク24の生成要求が発生した場合に（ステップS26）、同様に求められたフィルタB・タスク24の優先度は「20」だとすると（図7参照）、カーネルは優先度の高いフィルタB・タスク24を生成（実行）する。

【0014】本発明に係るマルチタスク制御方法の第2実施形態について図1に示すフローチャートを参照して説明する。

【0015】あるタスクのタスク生成（実行）要求があった場合（ステップS3）、カーネル1はタスク2に予測負荷情報の提供を依頼する。タスク2は該タスクの現時点に設定されているパラメータに対応する命令コードの数を予測負荷情報として（ステップS7）、カーネル1に渡す。カーネル1はタスク2から予測負荷情報を獲得すると（ステップS4）、その予測負荷情報とプロセッサのMIPS値とプロセッサの現時点での稼働状況

（例えば、ここでは、説明の簡単のため、MIPS値から求まるプロセッサが所定時間内に実行可能な命令コードの数（処理能力）に対する、現時点でマルチタスク処理で実行されている全てのタスクの命令コードの数の合計値（実測値を用いる場合と、予測負荷情報として各タスクから提示された命令コード数の合計値であってもよい）の割合）に基づいて、プロセッサの余力内でのタスク2が実行可能か否かを判断する（ステップS9）。このとき、カーネル1は、タスク2から受け取った予測不可情報としての命令コードの数をプロセッサのハードウェア構成に依存する修正を行ってもよい。例えば、タスク2から受け取った命令コードの数にプロセッサに依存するオーバーヘッド分を加算してもよい。タスク2の予測負荷情報とプロセッサの稼働状況とから、タスク2の命令コードの数がプロセッサの余力内で実行可能な命令コードの数であるとき、タスク2の実行可能と判断し、一方、プロセッサの余力内で実行可能な命令コードの数を越えているとき、タスク2の実行不可と判断する。実行可能なら（ステップS10）タスク2を生成（実行）する（ステップS6）。また、実行不可能な場合は（ステップS9）、タスク2の生成（実行）を拒否する（ステップS12）。

でそのタスクを実行可能か否かを予め判断できるので、プロセッサの処理能力を超えるマルチタスク処理を回避することができる。

【図面の簡単な説明】

【図 1】本発明に係るマルチタスク制御方法の第 2 の実施形態について説明するためのフローチャート。

【図 2】本発明に係るマルチタスク制御方法の第 1 の実施形態について説明するためのフローチャート。

【図 3】本発明に係るマルチタスク制御方法の第 3 の実施形態について説明するためのフローチャート。

【図 4】図 1、図 2 の処理動作を具体的に説明するための図。

【図 5】図 3 の処理動作を具体的に説明するための図。

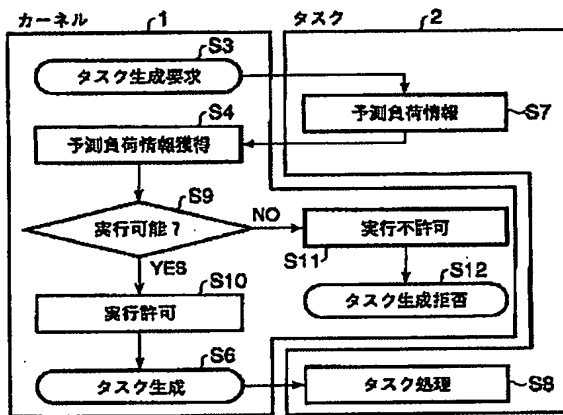
【図 6】図 1、図 3 の処理動作を実施の形態を説明するための説明図。

【図 7】図 2 の処理動作を具体的に説明するための図。

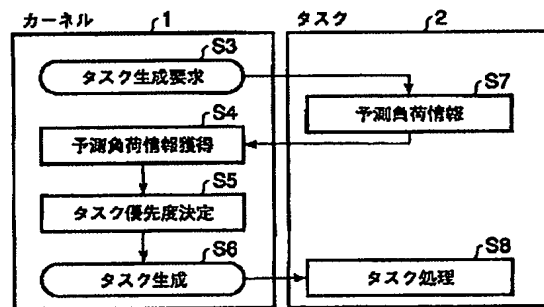
【符号の説明】

- 1…カーネル
- 2…タスク
- 20…プロセッサ
- 21…エンコーダ・タスク
- 22…デコーダ・タスク
- 23…フィルタ A・タスク
- 24…フィルタ B・タスク

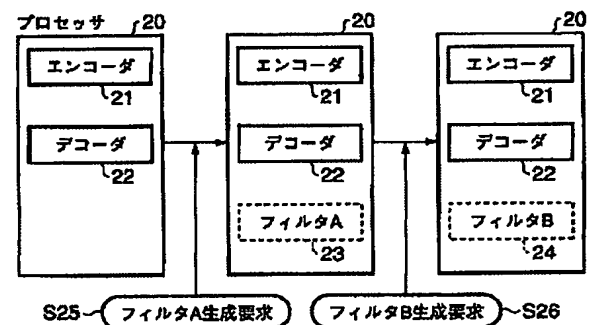
【図 1】



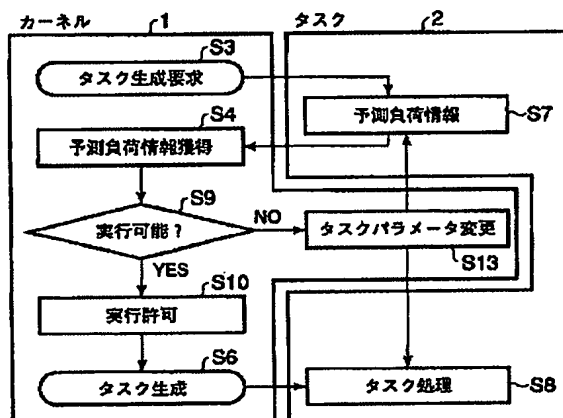
【図 2】



【図 4】



【図 3】



【図 6】

タスク	パラメータ	予測負荷
エンコーダ	1	40
デコーダ	1	40
フィルタA	1	20
	2	30
フィルタB	1	10

【図 7】

タスク	優先度
エンコーダ	実行中
デコーダ	実行中
フィルタA	10
フィルタB	20